Chapter 1

Introduction

Flow is everywhere: airflow around an airplane, a sailing ship, a windmill, or stirring of a cup of coffee to name some examples. Although it might not always be important to fully understand how the fluid is exactly flowing, there is a great demand for computer simulation of flow, called Computational Fluid Dynamics (CFD). One could think of the chemical industry (for example mixing applications), biomedical applications (blood flow through arteries, respiratory flow in the lungs), marine and off-shore engineering (for example resistance of ships or wave impact on vessels), and the automotive and aerospace industry (combustion in engines, climate control in the passenger compartment, aerodynamics).

In 1822, Navier derived the equations for the motion of fluid [33]. These equations are now known as the Navier-Stokes equations (Stokes independently rederived the equations in 1845 [42]). However, despite the existence of powerful computers, it is not possible, and it will not be possible in the near future, to compute most of the mentioned flow problems, as 'real life' flow is often too complicated. For instance, even in the calm situation where heated-up air from a cigarette rises, the smoke, visualising the air flow, reveals a complicated flow pattern full of details. This flow is hard to compute, but industrial applications are even more complicated.

Complexity of fluid flow is mainly determined by the combination of the flow medium, the flow velocity, and the objects involved in the flow geometry. Furthermore, other processes, like chemical reactions, thermal processes, or elastical/moving geometries further increase the complexity of the problem. These processes are not taken into account in this thesis.

In this thesis, a method is proposed for Direct Numerical Simulation (DNS) of turbulent flow in complex geometries. The flow domain is covered by a Cartesian grid, and boundaries that do not coincide with grid lines are handled by the cut-cell method. The approach is based on a 'symmetry-preserving'/'energy-conserving' discretisation [52, 53]. Before these terms are explained, first some mathematical background is given about turbulence and computational work in simulations.

1.1 Simulation of turbulent flow

1.1.1 Turbulence

Flow can be classified as laminar flow, where streamlines show a smooth, regular pattern, and turbulent flow, where flow particles seem to move criss-cross. Laminar flow is computationally relatively cheap to compute; the problem arises for turbulent flow. When simulating turbulent flow, the complex flow structures and irregular behaviour in time ask for a fine computational grid and small time steps. For most industrial applications, this computation is often not feasible. When a particular flow cannot be computed, it may be approximated. The solution can be found in the computation of the mean flow only. One is often not interested in the movements of small scales of motion, but only in the mean flow or even only in mean quantities (like drag and lift). To do this, the influence of the movements of small scales of motion on the large-scale flow needs to be modelled.

1.1.2 Direct Numerical Simulation

In general, three computational approaches exist for the computation of turbulent flow: Direct Numerical Simulation (DNS), Large Eddy Simulation (LES), and Reynolds Averaged Navier-Stokes (RANS). In practice, combinations of these three approaches are also used. In case of DNS, turbulence is not modelled and all details need to be resolved from the Navier-Stokes equations in the simulation; in the second case, a localised spatial filter is applied, which removes the small-scale details; and in the latter case, a long-time temporal filter is applied resulting in computation of the mean flow which is steady in time. The computational effort for LES lies in between that for DNS and RANS. For most industrial applications RANS is used. For both LES and RANS a closure model needs to be specified describing the influence of the turbulence on the computed flow. This can be rather difficult; different applications might require different turbulence models, and developing a well-performing turbulence model can be subject of research on its own.

Although the Navier-Stokes equations provide the appropriate model for the nonlinear dynamics of turbulence, most turbulent flows cannot be computed directly, because they possess far too many dynamically significant scales of motion. The multitude of small scales results from the convective term in the Navier-Stokes equations that allows for the transfer of energy from the large scales, at which the flow is driven, to the smallest scales that can survive dissipation. This range of dynamically significant scales of motion of a flow is determined by the Reynolds number (Re), which is a dimensionless number defined as

$$\mathrm{Re} = \frac{UL}{\nu},$$

where U is the typical speed of the flow, L is a typical length in the flow geometry (when flow around a sphere is examined for instance, it is the diameter of the sphere), and ν is the kinematic viscosity of the flow, which is a constant depending on the flow medium (ν is of the order $10^{-5}m^2/s$ and $10^{-6}m^2/s$ for air and water, respectively). The Reynolds number is a measure for the ratio between the effect of convection and diffusion; it is large for many flows. For a typical aerodynamic application (airflow around a flying airplane

for instance) the Reynolds number is around 10^7, while for hydrodynamic applications (large ships for instance) it is even 100 times higher.

When the computational grid needs to be fine enough to resolve the smallest relevant scales in isotropic turbulence, the number of grid cells per direction has to be of the order $\mathrm{Re}^{3/4}$ [21, 22]. Since space is three dimensional, the number of grid cells is proportial to $\mathrm{Re}^{9/4}$. But that is not all; the flow patterns in the smallest scales vary faster than the global flow. To track these movements in time, a small time step is required. This time step scales with $\mathrm{Re}^{1/2}$. Altogether, this means that the computational complexity scales with $\mathrm{Re}^{11/4}$. To indicate what this means, consider a CFD simulation that takes 1 hour. When the Reynolds number is increased with a factor 100, the time required for the simulation will be more than 36 years.

Although DNS can rarely be used for 'real applications', it is a very important method for several reasons:

- Since no terms in the flow equations are modelled, the correct answer is obtained as soon as the grid is fine enough and the time step small enough (once one is sure that the numerical method works fine and the program contains no errors); hence DNS can give basis data for turbulence research.

- When the small-scale flow is important, because it introduces sound or vibrations for instance, or the flow is forced at the small scales, *e.g.*, by chemical reactions (combustion), DNS is a method that can predict this correctly.

- Methods for DNS can form the basis for RANS and LES and are often used for tuning of unknown constants in the turbulence model (for a test case at a low Reynolds number).

1.1.3 Symmetry and conservation

The continuity equation describes the conservation of mass. The conservation of momentum is expressed by the Navier-Stokes equations in terms of convection, diffusion, pressure, and external forces. The latter are left out of consideration here. From the continuity equation and Navier-Stokes equations an equation for the evolution of the total kinetic energy of the flow can be formed. This equation does not contain convective and pressure terms (except on boundaries of the domain); the dynamics of the total energy of the flow is governed by a diffusive equation. Mathematically, this is an immediate consequence of the following symmetry properties of the differential operators in the Navier-Stokes equations:

- The convective operator is skew symmetric.

- Diffusion is a symmetric, negative definite operator.

- The gradient operator is minus the adjoint of the divergence operator. This implies that pressure gradient will not change the total kinetic energy.

For the following reasons it is important that the numerical approximations of the convective, diffusive, and pressure terms mimic these properties:

- For slightly viscous flow, the contribution of diffusion seems negligible compared to convection. But without 'that little bit' of diffusion one would not encounter head wind on a bicycle and airplanes will not have lift. It was only found in 1903 by Prandtl, by visualising a boundary layer, that diffusion does play an essential role.

- Flow is built from different length scales. Usually, boundary conditions drive the flow at the largest scale. From there, energy is transferred by convection to smaller scales and so on, until the smallest scales of motion. At this scale, diffusion is most efficient and prevents the flow from breaking up into smaller scales of motion. The scales in between the largest and smallest scale form the inertial range. It is rather independent of the boundary conditions on the one hand and diffusion on the other. The balance in this subtle energy cascade can only be captured well when the roles of convection and diffusion are strictly separated.

- Last but not least, numerical simulation often suffers from instabilities. These can be caused by grids that are too coarse, abrupt changes in grid sizes, or lack of dissipation (diffusion). When this happens, velocities grow unboundedly and the simulation will break down. Since a large negative velocity can compensate a large positive velocity, the total amount of momentum is not affected, but the total amount of energy does increase. So conservation of kinetic energy guarantees stability, irrespective of dissipation and the (maybe awfully chosen) grid (as long as a stable time discretisation is provided).

1.2 Numerical model

1.2.1 Overview of numerical methods

For solving a flow problem, the Navier-Stokes equations and the continuity equation (describing conservation of mass) have to be solved simultaneously. This is done by discretising this set of equations, which in principle can be done by any of the following methods:

- spectral method;

- finite element method;

- finite difference method;

- finite volume method.

In the 80's, *spectral methods* were the prevailing approach for DNS. Their strength is in the high asymptotic order, but when the computational domain contains corners, this efficiency is dramatically decreased, since the solution is described by globally defined smooth functions. Therefore, spectral methods are only used in relatively simple geometries. In more complex geometries the spectral element method can be used to solve these problems by restricting the basic functions to only a part of the flow domain. More details about spectral methods can be found in [6] and [25] for instance.

The *finite element method* is most commonly used in combination with unstructured grids. An unstructured grid is constructed by dividing the geometry into small cells that can have different shapes. In combination with a chosen basis function such a cell is called an element. The approximate solution is built by combining all these functions. The lack of structure makes simple computations more time (and memory) consuming than on a structured grid with the same number of cells [14]. This is probably the reason that this method is not so often used for DNS; the strength of unstructured grids lies in its flexibility for very complex geometries.

The *finite difference* method discretises a differential equation to a difference equation. It is especially suited for (curvilinear) structured grids where unknowns are all aligned with each other. The accuracy can be increased where this is required. A disadvantage is that the conservation form the Navier-Stokes equation is numerically not maintained such that the total momentum is usually not conserved. In [31] for example the finite difference method on a curvilinear grid is used for a similar problem as is presented in this thesis.

The *finite volume method*, which is the method used in this thesis, discretises a mathematical model from its conservation form. Although the Navier-Stokes and continuity equations are often shown as differential equations, they are actually physical conservation laws, which give the balance of momentum (and mass) that holds for any volume. If the conservation laws are applied to an arbitrarily small volume around an arbitrary point in the flow domain, the differential equations are, provided a sufficient smooth solution, obtained. When applying the finite volume method to the Navier-Stokes equations, the geometry has to be covered with cells; as for the finite element method, this can be a structured as well as an unstructured grid. Since structured grids are more efficient in a computer program, these grids are preferred for DNS. When using a structured grid, usually a curvilinear grid is chosen, see [14, 13] for example, but in this thesis a Cartesian grid is used. Without going into detail, discretising in a finite volume manner means that for each cell locally the conservation law is applied, *e.g.*, for momentum. For each time step, the inflow of momentum through the cell boundary is computed, with contributions from convection, diffusion, and pressure differences. When the net inflow through the cell boundary is positive, the momentum in the cell increases which means that the velocity increases. A cell face is always part of the boundary of its two adjacent cells; when the flux through the face is equal for both cells, then the (numerical) momentum is exactly conserved.

1.2.2 Symmetry-preserving finite volume discretisation

Based on the considerations described above, a symmetry-preserving discretisation was developed at the University of Groningen [52, 53]. The following properties of the discretisation are obtained, which correspond to properties of the differential equations:

- The convective operator is skew symmetric. This implies that convection does not contribute to the total kinetic energy.
- Diffusion is a symmetric, negative definite[1] operator. This means that it will dissipate energy from any non-zero velocity field.

[1] semi-definite is avoided by assuming that the boundary conditions are such that no eigenvalue equal

- The discrete gradient operator is minus the adjoint of the discrete divergence operator. This implies that the pressure gradient will not change the total kinetic energy.

In this way, the numerical operators have the same symmetry properties as their analytical counterparts. The convective operator has purely imaginary eigenvalues due to its skew symmetry and the diffusive operator has real, negative eigenvalues. As a consequence, the eigenvalues of their combination all lie in the stable half-plane, which makes the spatially discretised system stable.

This discretisation was originally developed for Cartesian grids with grid-aligned geometries. This limits its applicability, but nevertheless the power of energy conserving discretisation was proven during several workshops. For instance, flow around a square cylinder has been a test case in two workshops for LES/DNS methods, see [39, 50]. The results showed a wide spread in the drag coefficient. Remarkable is that all upwind discretisations (which add artificial viscosity) had a too high drag, whereas central discretisations were centred around the experimental data. The result of the symmetry-preserving discretisation was, together with two other simulations, within the range of measurements. It was concluded at the workshops that this test case forms a major challenge to the LES techniques.

In 1997, one of the test cases at the 6^{th} ERCOFTAC Workshop was flow in a channel with surface mounted cubes. The Reynolds number was $13,000$ based on the channel width and mean bulk velocity. Four groups presented their results of RANS simulations, there were no results of Large Eddy Simulations, and DNS results of the symmetry-preserving discretisation were presented. On a similar grid as was used for the RANS simulation that agreed best with the available experimental data, better results were obtained with the symmetry-preserving DNS method [47, 51].

1.2.3 Cartesian grid methods for flows in complex geometries

To be able to deal with arbitrary geometries, the discretisation on Cartesian grids has to be extended to deal with complex geometries. Generally two possibilities exist:

- the grid aligns with the geometry;
- the grid does not align with the geometry.

In the first case, a restriction is made on the geometry. For instance, one can compute the flow around a cylinder with a rectangular cross section, but not with a circular cross section on a Cartesian grid. In the latter case, the physical geometry could be approximated by using a stair-case geometry, where cells on the cylinder boundary are either completely open for flow, or completely closed. Although this approximation gets better when the grid is refined, the normal will always point in the horizontal or vertical direction, causing a non-smooth pressure and velocity profile along the cylinder surface. This influences physical properties like flow separation.

These 'stair-case problems' can generally be solved in two different ways. One way is to use an immersed boundary technique. In this case, no internal objects exist with

to zero exists, *e.g.*, by assuming that at least a part of the boundary of the flow domain consists of a no-slip solid boundary

solid boundaries, but everything consists of flow media. A simple Cartesian grid is used that does not conform to the shape of the body in the flow. Instead of implying a boundary condition, an additional term, called the forcing function, is added to the governing equations. With the right forcing function the desired boundary condition can (approximately) be obtained. Often, this makes the virtual fluid-solid transition somewhat smooth. Examples of an immersed boundary method can be found in [26, 46].

The other way is by using cut cells, which are cells that are partially open for flow. Now, cells no longer appear block-shaped, which introduces a new difficulty. The main part of the grid will keep the simple form of the Cartesian grid, but cells near the boundary require more attention. Their shapes are irregular, the size can be small, and the locations where the discrete velocities and pressure are defined do not need to be aligned anymore. A significant part of this thesis is dedicated to the extension from Cartesian grids with grid-aligned boundaries to Cartesian grids with cut cells.

In addition to the problems with cut cells described above, another difficulty is to retain the symmetry properties in the cut cells. Normally, when using explicit time integration, the maximum time step is restricted for both the convective and the diffusive operator. This restriction depends on the cell sizes: when the cell gets smaller, the maximum time step gets smaller. If no lower limit exists for the cell size, the maximum time step might get unacceptably small. This is the case when using cut cells, since these cells could be very small. For the symmetry-preserving convective discretisation, however, the time step restriction is not sharpened for cut cells, although it normally depends linearly on the cell sizes. It is shown that the skew symmetry is preserved in a trivial manner for cut cells, and, even more, it is proven that the eigenvalues of the skew-symmetric convective operator remain bounded in terms of the uncut cell sizes. However, this property does not hold for diffusion, where the time step restriction depends on the square of the cell sizes. A new proposal is introduced for the discretisation of diffusion in cut cells that does possess the required properties. The method is based on a combination of two discretisations, one that does not have the right stability properties, but is accurate, and one that is always stable, but is less accurate for cut cells with non-vanishing cell volume. The combination is made such that the magnitude of the eigenvalues does not depend on the cell sizes. Combining this, a symmetry-preserving discretisation is found for a Cartesian grid with cut cells. No additional time step restrictions arise for the cut cells.

1.2.4 Verification and validation

The new discretisation has to be tested with various test cases. First, the discretisation in cut cells is tested with flow through a channel. Normally, this is one of the most simple flows, but when the channel is placed skew to the grid, it is a challenging test case for a discretisation with cut cells. At certain inclination angles all types of cut cells will occur and from a mathematical point of view, this simple geometry is the most complex geometry with straight boundaries that can occur.

Next, simulations are performed in a geometry with curved boundaries to show the behaviour of the numerical method for curved boundaries. A test case is chosen that is relatively simple, but nevertheless non-trivial: flow through a bent channel. Again, all types of cut cells occur, but the solution is physically simpler than flow around a circular

cylinder where separation can occur.

After obtaining the results for these simple test cases, results are presented for the flow around circular cylinders at three different Reynolds numbers. At the lowest Reynolds number, the flow is still laminar and steady. The dependence of the solution on various parameters is studied. This is necessary for the higher Reynolds numbers, where de solution is more complicated resulting in longer time required for the simulation. The highest Reynolds number is a challenging turbulent flow simulation for a numerical method without grid aligned boundaries, and the results will be compared with available results from measurements and many other simulations.

1.3 Outline

In this thesis, the energy conserving discretisation is extended from Cartesian grids with rectangular cells to Cartesian grids with cut cells. This thesis consists of six chapters including this introduction.

In Chapter 2 the Navier-Stokes equations are presented including the boundary conditions. Importantly, the key to conservation of the kinetic energy is introduced: the equation for conservation of kinetic energy is derived from the Navier-Stokes equations. Attention is paid to the symmetry properties the different (analytical) operators have, since this will play the central role in the derivation of the discretisation.

Chapter 3 presents the numerical model, *i.e.*, the discretisation of the mathematical model. To start, the choice for the Cartesian grid is explained. Then, the location of the unknowns is given and, in particular, the staggered grid is explained. Hereafter, the construction of cut cells is illustrated. The conservation properties of kinetic energy are applied for a general discrete model. This establishes important properties for the discrete operators (divergence, convection, diffusion, and gradient, respectively). These properties form the base for the spatial discretisation of the convective, diffusive, divergence, and gradient operator that are derived next. Care is taken that the stability properties are preserved in the cut cells. Finally, the temporal discretisation is introduced and the evolution of energy is shown for the completely discretised system.

In Chapter 4, the results of some relatively simple test cases are shown to support some of the choices made in the discretisation, as well as to illustrate the performance of the discretisation in cut cells. The simplest test case is a channel placed skew to the grid. To test curved boundaries, flow along a curved boundary is studied.

Chapter 5 presents the results of flow around a circular cylinder at different Reynolds numbers, including results for Re = 3900. DNS of this flow is computationally expensive; only one other DNS at this Reynolds number was found, but many Large Eddy Simulations and experiments exist to compare with.

Finally, Chapter 6 contains the summaries and conclusions.